

# HP kalkulatori i UserRPL

Saša Vranić\*

**SAŽETAK.** U ovom radu ukratko su objašnjeni HP kalkulatori i UserRPL. UserRPL je programski jezik koji se koristi kod pisanja jednostavnijih programa za HP kalkulatore. U radu je navedena kratka povijest i vrste HP kalkulatora te objašnjena logika pisanja programa, operacije za manipuliranje podacima u stogu, uvjetne strukture, petlje, načini dobivanja ulaznih, te prikazivanja izlaznih podataka.

**KLJUČNE RIJEČI:** HP kalkulator, RPL, UserRPL, RPN

## 1. Uvod

Motivacija pisanja ovoga rada je bila činjenica da je na Geodetskom fakultetu postoji mnogo studenata koji kod polaganja ispita kao što su Teorija pogrešaka (AIOGM) ili Državna izmjera koristi HP kalkulatore, ali znaju vrlo malo o mogućnostima tih kalkulatora. Za otprilike 1500 kn koliko košta najnoviji model (HP-50G) mogli bismo očekivati da će kalkulator sam riješiti postavljeni zadatak. No činjenica je da ukoliko ne znamo kako funkcionira taj kalkulator i koje su njegove mogućnosti, rezultat neće biti zadovoljavajući.

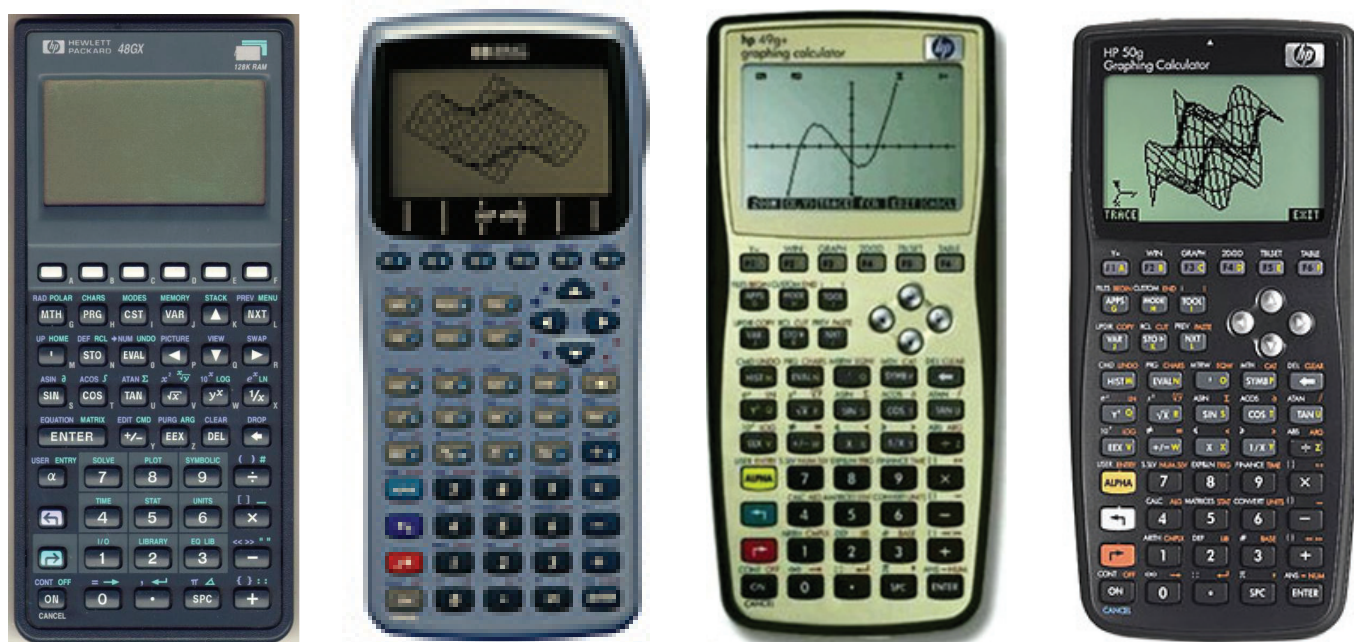
HP-48 je serija grafičkih kalkulatora koja uključuje modele HP-48S, HP-48SX, HP-48G, HP-48GX (Slika 1) i HP-48G+. Modeli G su poboljšane verzije S modela, a razlike između pojedinih G modela je u količini memorije koju posjeduju.

Sljedeća serija kalkulatora je HP-49. Prvi kalkulator u toj seriji bio je HP-49G

(Slika 1) koji je uključio mnoge alate iz HP-48 serije. HP-49G je bio prvi kalkulator koji je koristio flash memoriju i ROM memoriju koja se može nadograditi. HP-49G+ (Slika 1) je predstavljen 2003. godine, te su uvedene mnoge promjene od kojih navodimo neke: fizički izgled, nova arhitektura procesora (ARM umjesto dosadašnje SATURN), utor za SD kartice (do 2 GB), komunikacija putem USB-a i IrDA. Nova arhitektura procesora ARM rezultirala je bržim izvođenjem zadataka (3-7 puta brže nego na HP49G, ovisno o zadatku). Sljedeći i zasada posljednji kalkulator u ovoj seriji je HP-50G (Slika 1), predstavljen 2006. godine, koji ne donosi neka bitna poboljšanja. Neka od najočitijih su: drugačija boja kalkulatora, poboljšane tipke, 4 AAA baterije umjesto dosadašnjih 3. Osim ovih kalkulatora, postoji još jedan kalkulator HP-48G II koji nije kao što mu ime sugerira iz serije HP-48. To je kalkulator sa istim mogućnostima kao i HP-49G+, ali bez utora za SD karti-

cu i s manje unutarnje memorije.

Svi navedeni kalkulatori imaju oko 2300 ugrađenih funkcija. Osim tih funkcija korisnik može sam definirati vlastite programe. Za to postoje dva načina, odnosno dva programska jezika, UserRPL i SystemRPL. RPL (engl. ROM-based Procedural Language ili Reverse Polish Lisp) je programski jezik razvijen od strane programera iz HP-a i korišten kod HP kalkulatora. UserRPL je varijanta RPL-a koja je jednostavnija i sigurnija od SystemRPL-a. Kod pisanja programa u UserRPL-u kalkulator provjerava da li su u stog (engl. stack) uneseni svi potrebni argumenti. Ako nisu, javlja pogrešku i zaustavlja izvođenje. SystemRPL uključuje pristup većem broju funkcija i objekata. Kod pisanja programa u SystemRPL-u potrebno obratiti pažnju jer se ulazni podaci ne provjeravaju pa i najmanja pogreška može dovesti do pada sustava. Programi napisani SystemRPL-u se zbog toga izvođe mnogo brže od programa napisanih u UserRPL-u.



Slika 1. HP kalkulatori (HP-48GX, HP-49G, HP-49G+, HP-50G)

\* Saša Vranić, usmjerenje: Inženjerska geodezija i upravljanje prostornim informacijama, Geodetski fakultet Sveučilišta u Zagrebu, e-mail: [svranic@geof.hr](mailto:svranic@geof.hr)

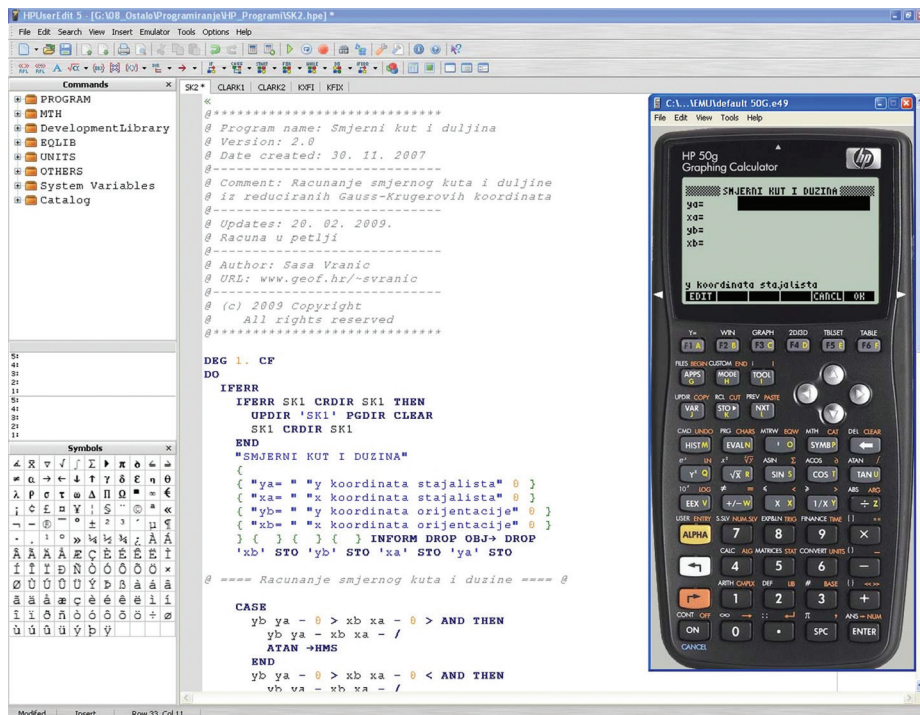
U nastavku ćemo ukratko objasniti način pisanja programa u UserRPL-u. Pritom ćemo se ograničiti na pisanje programa za kalkulator iz serije HP-48 i HP-49. Programe možemo pisati pomoću programskog paketa HPUserEdit 5 (Slika 2) ili direktno u kalkulatoru.

## 2. Osnovni koncepti

Kod HP kalkulatora postoji RPN i algebarski način rada. Kod algebarskog načina rada izrazi se pišu uobičajenom matematičkom notacijom. RPN (engl. Reverse Polish Notation) je način pisanja matematičkih izraza gdje se prvo upišu operandi, a zatim se upisuje operator (npr. da bi zbrojili dva broja 2 i 3, pišemo 2 3 +; ili ako želimo izračunati sinus kuta, prvo upišemo argument funkcije pa onda funkciju). Stog (engl. stack) je prostor na zaslonu u kojem manipuliramo objektima kod RPN načina rada (Slika 3). Broj razina stoga koji se prikazuje na zaslonu ovisi od veličine fonta (između 5 i 10), a maksimalni broj razina koji može biti ispisan je ograničen memorijom kalkulatora.

Objekti mogu biti pohranjeni u memoriju na dva načina. Možemo ih pohraniti kao lokalne ili globalne varijable.

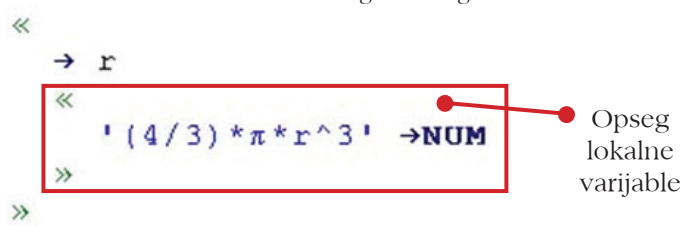
Objekti pohranjeni u lokalne varijable su dostupne privremeno, tj. samo za vrijeme izvođenja programa. Nakon izvođenja programa, sve lokalne varijable se brišu (Slika 4). U ovom primjeru, program uzima broj s prve razine stoga



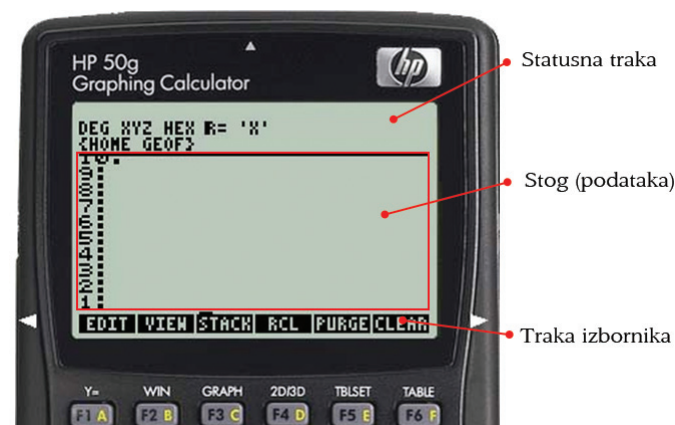
Slika 2. Sučelje programskog paketa HPUserEdit 5

(dakle, argument moramo unijeti prije pogretanja programa), sprema ju u lokalnu varijablu r i s njom računa volumen sfere. Naredba -> NUM se koristi za evaluiranje algebarskog izraza.

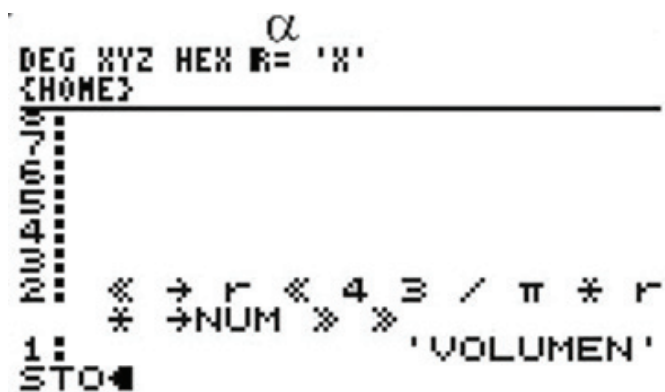
Globalne varijable su spremne trajno u memoriju kalkulatora i možemo ih vidjeti pritiskom na tipku VAR. Prethodni



Slika 4. Primjer korištenja lokalne varijable



Slika 3. Zaslon kalkulatora HP-50G



Slika 5. Pohrana globalne varijable

Tablica 1. Tipovi objekata kod HP kalkulatora

Objekt	Primjer
Broj	1
String	"A"
Program	<< 2 3 + >>
Lista	{ 1 2 3 }
Vektor	[ 10 11 12 ]
Matrica (polje)	[[1 2 3][4 5 6]]
Algebarski izraz	'23+34'

Tablica 2. Operacije za manipulaciju objektima u stogu

Naredba	Učinak
CLEAR	Čisti čitav stog
DROP	Briše objekt na prvoj razini stoga
SWAP	Zamjenjuje objekte na 1. i 2. razinu
ROLL	Pomiče navedenu razinu na prvu razinu (npr. 4 ROLL pomiče objekt na 4. razini na 1. razinu)
DUP	Kopira objekt na prvoj razini
DUPN	Kopira objekte na prvih n razina (npr. 6 DUPN će kopirati prvih 6 razina)
Algebarski izraz	'23+34'

jednostavnijih programa (time osiguravamo maksimalno brzo i jednostavno izvođenje programa). Postoji 19 funkcija koje omogućavaju kopiranje, brisanje, premještanje objekata unutar stoga. U nastavku ćemo prikazati nekoliko najčešće korištenih.

Način rada kalkulatora određen je sistemске zastavice (engl. system flags). Možemo ih shvatiti kao skup varijabli koje mogu poprimiti vrijednosti uključeno/isključeno ili 1/0. Imena sistemskih zastavica su oblika negativnih brojeva (-1 do -128). Npr. ako je sistemška zastavica -3 postavljena uključen je numerički mod, tj. simbolički argumenti i konstante (a, b,  $\pi$  i sl) se zamjenjuju njihovim numeričkim vrijednostima. Pored sistemskih zastavica postoje i korisničke zastavice, koje su oblika pozitivnih brojeva (od 1 do 128).

### 3. Uvjetne strukture i petlje

Uvjetne strukture i petlje kao i kod svakog programskog jezika su konstrukcije koje dodavaju fleksibilnost programu. U nastavku ćemo ukratko objasniti te konstrukcije.

Uvjetne strukture provjeravaju uvjet i ovisno o tome da li je uvjet ispunjen ili nije izvršavaju određeni dio koda. Osnovne uvjetne strukture su IF ... THEN ... ELSE i CASE (Slika 6). Kod postavljanja uvjeta se naravno mogu koristiti operatori AND, NOT, OR, XOR).

Petlje omogućavaju ponavljanje izvođenja koda određeni broj puta. Pritom imamo konačne (koje se ponavljaju određeni broj puta) i beskonačne petlje (koje se ponavljaju neodređeni broj puta). U nastavku ćemo dati primjer za svaku od ove dvije vrste petlje.

Primjer konačne petlje je FOR ... NEXT (Slika 7, lijevo). Ovdje je napisan primjer koji računa sumu prvih 20 brojeva. U primjeru varijable a i b su početna i završna vrijednost pomoćne varijable i. Nakon što se izvrši blok koda unutar petlje, dolazimo do ključne riječi NEXT koja brojač petlji i povećava za jedan i testira vrijednost. Ako je jednaka 20, petlja se završava, u suprotnome se ponovno izvodi.

Isti program možemo napisati i pomoću beskonačne petlje WHILE (Slika 7, desno). Ovdje ne moramo navoditi broj iteracija koji će se izvršiti, već samo navedemo uvjet koji petlja mora zadovoljiti. Unutar petlje pojavljuje se naredba STO+. Ona zahtijeva dva argumenta, varijablu i broj ili vrijednost bilo kojeg drugog objekta, koje zatim zbraja i pohranjuje u navedenu varijablu (kao npr. naredba ++i u C++).

```

«      «
→ a b      → a b
«      «
      IF a b > THEN      CASE a b > THEN
        a b - √          a b - √
      ELSE              END
        "Pogreska: a < !  a b < THEN
      ELSE              "Pogreska: a < b"
        "a == b"          END
      END              "a == b"
«      «              END
»      »              »
»      »              »

```

Slika 6. Primjer uvjetne strukture IF...THEN...ELSE i CASE

```

« 1 20      «
→ a b      0 'sum' STO
«      0 'n' STO
      0 @ pocetna suma  WHILE n 20 ≤
      a b FOR i          REPEAT
        i +              'sum' n STO+
      NEXT              'n' 1 STO+
«      »              END
»      »              »

```

Slika 7. Petlje

Osim ovih petlji postoje i druge koje funkcioniraju vrlo slično, uz vrlo male razlike. Konačne petlje START ... NEXT i FOR ... STEP te beskonačna petlja DO ... UNTIL.

### 4. Dobivanje ulaznih podataka

Postoji više načina dobivanja ulaznih podataka. U dosadašnjim primjerima je bilo potrebno znati koje argumente program zahtijeva i kojim redoslijedom. Ovdje ćemo objasniti dvije funkcije, INFORM i CHOOSE koje omogućavaju unos podataka bez poznavanja načina na koji funkcionira program. Osim ove dvije funkcije postoje i još neke koje nećemo objašnjavati.

Naredbom INFORM kreiramo obrazac u koji onda unosimo ulazne podatke (Slika 8). Prije naredbe INFORM idu njezini argumenti (pošto radimo u RPN

modu), a iza nje ide naredba DROP. Naime, naredba INFORM u slučaju unesenih svih argumenata ispisuje 1 na prvu razinu stoga, u suprotnome ispisuje 0. Na drugu razinu ispisuje upisane vrijednosti u obrazac u obliku objekta tj. liste. Naredbom OBJ-> elemente liste razdvajamo u zasebne brojeve i time omogućavamo da svaku vrijednost pohranimo u zasebnu varijablu. Ta naredba također vraća na prvu razinu stoga 1, pa moramo upotrijebiti naredbu DROP. U nastavku programa svaka vrijednost se pohranjuje u lokalnu varijablu i računa se udaljenost između točaka A i B.

Naredba CHOOSE omogućava kreiranje izbornika (Slika 9). U sljedećem primjeru smo napravili izbornik koji predstavlja prečac do već napravljenih programa. Prvi redak sadrži string koji predstavlja naslov izbornika, zatim slijedi lista stavki izbornika. Svaka stavka (redak) se sastoji od stringa

```

«      «
"SMJERNI KUT I DUZINA"  SMJERNI KUT I DUZINA
{
  { "y" "y koordinata tocke A" 0 }  y=
  { "xa" "x koordinata tocke A" 0 }  xa=
  { "yb" "y koordinata tocke B" 0 }  yb=
  { "xb" "x koordinata tocke B" 0 }  xb=
} { } { } { } INFORM DROP OBJ-> DROE
→ ya xa yb xb
«      »
      yb ya - SQ xb xa - SQ + √
»      »
      y koordinata tocke A
      EDIT  CANCEL OK

```

Slika 8. Naredba INFORM



```

«
  ***Drzavna izmjera***
{
  { "Razlike sirina" « DI RAZL » }
  { "Racunanje b iterativno" « DI BITE » }
  { "Radijus zakr. M,N,Ra" « DI RADMN » }
  { "Radijus kosog presjeka" « DI RADKP » }
  { "Duzina luka meridijana" « DI DUZM » }
  { "Duzina luka paralele" « DI DUZN » }
  { "Konverzija: fi,la,h→XYZ" « DI KFIX » }
  { "Konverzija: XYZ→fi,la,h-dir" « DI KXF » }
}
1 CLLCD
IF CHOOSE THEN
  EVAL
END
»

```

Slika 9. Naredba CHOOSE



ga koji se prikazuje u izborniku i programa koji treba pokrenuti. Iza liste je broj 1 koji predstavlja indeks početno izabrane stavke. Naredba CLLCD (Clear LCD) čisti zaslon da bi se mogao prikazati potprogram koji se pokreće. Naredba CHOOSE također vraća 1 ako je izabrana stavka s izbornika, u suprotnome vraća 0.

## 5. Prikazivanje izlaznih podataka

Bez korištenja funkcija za prikaz izlaznih podataka, izlazni podaci bi bili jednostavno vraćeni u stog. Objasniti ćemo dvije funkcije, MSGBOX i DISP. MSGBOX (Slika 10) je funkcija prikladna za prikaz upozorenja (slično funkciji alert u npr. Javascriptu).

Funkcija DISP (Slika 11) je prikladna za prikaz stringova. Svi ostali objekti će biti prilikom prikazivanja pretvoreni automatski u stringove.

Na prethodnoj slici je prikazan dio koda programa koji računa smjerni kut i dužinu. U prvom redu imamo varijablu koja sadrži numeričku vrijednost smjernog kuta, zatim funkciju →STR koja pretvara taj broj u string, zatim imamo sljedeći string koji opisuje tu vrijednost. Te vrijednosti se sada nalaze u stogu i naredbom SWAP im zamijenimo redoslijed. Znakom + ta dva stringa spojimo u jedan. Naredbom CLLCD očistimo zaslon, tako da nema ništa u pozadini i na kraju upotrijebimo naredbu DISP. Prvi argument je na drugoj razini stoga (string koji se prikazuje), a drugi argument je cijeli broj koji predstavlja broj linije na kojoj će se string prikazati (prvi string se prikazuje na trećoj liniji). Ovdje treba paziti na to koliko imamo dostupnih redaka stoga. Na kraju imamo dvije nove naredbe FREEZE i WAIT. Naredba FREEZE zamrzava dio zaslona tako da se prikaz ne ažurira dok se ne pritisne tipka. Koji će dio zaslona

```

«
  "Ovo je upozorenje" MSGBOX
»

```

Slika 10. Naredba MSGBOX



```

Smj.kut= 39.4820055932
N1a →STR "Smj.kut= " SWAP + Duzina = 390.512483795
3 DISP
d →STR "Duzina = " SWAP +
5 DISP
"Pritisnite ENTER za nastav

```

Slika 11. Naredba DISP



biti zamrznut ovisi od cijelog broja koji je argument te funkcije. U primjeru je to broj 3 koji definira da će se zamrznuti traka statusa i stog. Naredba WAIT zaustavlja izvođenje programa na neko vrijeme ili dok se ne pritisne tipka, ovisno o argumentu. U primjeru je dan broj 0, što znači da će izvođenje biti zaustavljeno dok se ne pritisne bilo koja tipka.

## 6. Zaključak

U ovom radu objašnjeni su samo osnovni koncepti programiranja u UserR-PL-u. Navedeni HP kalkulatori pružaju mnogo više. U nastavku ćemo navesti neke. Upotreba mnoštva ugrađenih funkcija (matematičkih, statističkih, funkcija za manipuliranje memorijom, varijablama, direktorijima), kreiranje privremenih izbornika u traci izbornika, crtanje po zaslonu. Kada bi htjeli objasniti sve ove naredbe i navesti praktične primjere ovaj rad bi bio znatno duži.

Tko želi saznati više o ovoj tematici, na webu ima mnoštvo materijala za učenje. Velika većina tih materijala je be-

splatna. Navedeni program HPUserEdit (URL-6) je također besplatan. Primjeri korišteni u ovom radu dostupne su na autorovoj web stranici.

## Literatura

- Hewlett-Packard Company (2003): HP 49g+ graphing calculator (user's guide), Hewlett-Packard Company
- Hewlett-Packard Company (2006): HP 49g+/ HP 48gII graphing calculator (advanced user's reference manual), Hewlett-Packard Company
- URL-1: <http://en.wikipedia.org/wiki/>, Wikipedia, 23. 02. 2009.
- URL-2: <http://www.hpmuseum.org/>, The museum of HP calculators, 23. 02. 2009.
- URL-3: <http://www.educalc.net/>, Educalc network, 23. 02. 2009.
- URL-4: <http://www.hpcalc.org/>, Hpcalc.org, 23. 02. 2009.
- URL-5: <http://www.hp.com/>, Hewlett-Packard Company, 23. 02. 2009.
- URL-6: <http://www.gaak.org/hpuser/>, HP UserEdit, 25. 02. 2009.